

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/795,923	Confirmation No.	1818
Inventor	:	Kegel		
Filed	:	03/08/2004		
TC/A.U.	:	2443		
Examiner	:	Jude Jean-Gilles		
Docket No.	:	I004-P03074US		
Customer No.	:	33356		

REPLY BRIEF

The following REPLY BRIEF is submitted in response to the corrected Examiner's Answer mailed November 13, 2008 for consideration by the Board of Appeals and Interferences. 37 C.F.R. § 41.41.

REPLY

A. The References May Not Properly Be Combined

The Examiner provides a motivation to combine Averbuj with Smith based on the lack of Smith teaching protocol engine limitations. Specifically, Claims 1 and 11 recite "invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command". And claims 21, 26 and 31 recite "each protocol engine to execute a command included in one of the scripts". The Examiner states that these limitation are not disclosed by Smith, but that they are disclosed by Averbuj. (Examiner's Answer, p. 4) However, we assert that the Examiner provided motivation to combine is improper.

The Examiner asserts that "allowing a variety of test algorithms to easily be defined and maintained centrally in the form of generalized commands, thereby eliminating the need to store common test algorithms in a distributed fashion" is the motivation to combine. (See Examiner's Answer, pp. 4-5) But this statement is wholly irrelevant to the protocol engine limitations. Here's why.

The sequencers in Averbuj receive a complete BIST algorithm and execute the algorithm one command at a time. (Averbuj, para. 0030) In contrast, the protocol engines as claimed receive and execute a single command from the test script or scripts. This differs from Averbuj which discloses sequencers that receive and execute an entire algorithm. The sequencers of Averbuj are arguably similar to the claimed script interpreters that they receive a whole script, but they are also unlike the claimed script interpreters in that the sequencers execute the scripts (that is, algorithms).

That the sequencers of Averbuj receive and execute an entire algorithm of multiple commands while the claimed protocol engine receives single commands eviscerates the motivation to combine provided by the Examiner. The “allowing a variety of test algorithms to easily be defined and maintained centrally in the form of generalized commands, thereby eliminating the need to store common test algorithms in a distributed fashion” is wholly irrelevant to the protocol engines that receive a command to execute. (See Examiner’s Answer, pp. 4-5) The protocol engines each receive and execute a command that is part of a test script. (Claims 1, 11, 21, 26 and 31) The protocol engines neither receive nor store an entire test algorithm. Therefore, the motivation to combine cited by the Examiner is inapplicable. As the cited motivation is inapplicable to the claimed protocol engines, there is no reason to combine the cited references.

Moreover, simply, Smith discloses a telecommunications testing system and Averbuj discloses a memory testing system. These publications disclose different techniques that solve problems that are wholly unrelated to one another. The low level, chip based testing techniques of Averbuj are not applicable to the higher level, inter-device and telecommunications network testing techniques taught in Smith. Smith and Averbuj are in different technology spaces. As such, their teachings may not be properly combined.

We reiterate that it is not apparent why a person involved with a telecommunications testing system of Smith would avail himself of the memory testing system of Averbuj. We maintain our assertion that “the Examiner has not provided a sufficient reason or explicit analysis of why the disclosures of the references should be combined.” *Ex parte Erkey et al.*, Appeal 20071375 (BPAI May 11, 2007).

We assert that there is no tie between the memory testing system of Averbuj and the telecommunications testing system of Smith. This is because Averbuj fails to mention telecommunications and fails to mention network protocols that are at the heart of the disclosure of Smith. And Smith fails to mention a Built-In Self Test (BIST) Architecture or an algorithm for testing memory modules in a single device that are at the heart of the disclosure of Averbuj.

B. The References Do Not Disclose The Claimed “Protocol Engine”

The Examiner asserts that the sequencers of Averbuj that conform to a command protocol to test memory devices disclose the claimed protocol engines. However, the command protocol of Averbuj is not equivalent to and does not teach the claimed protocol engine.

This is because the word protocol has multiple meanings. We note that “Claim terms must be interpreted, however, consistently with their use in the supporting specification”. *Ex parte Anderson et al.* Appeal 20084644 (BPAI December 22, 2008). The term protocol is used in both the pending application and in Averbuj. But the term is used in different ways in the pending application and in Averbuj.

A protocol may [1] be a “form of ceremony and etiquette observed by diplomats and heads of state”; [2] define a set of steps taken to test a medication or perform a medical test; [3] define a series of steps taken to achieve a scientific or computer generated test; or [4] define “rules determining the format and transmission of data”. Applicants have used the fourth definition in their specification and in their claims. Averbuj has used the third definition. Therefore, Averbuj cannot be cited as teaching the claimed protocol engine.

Simply, the claims recite a protocol engine that involves network traffic which the specification defines as “data units communicated over a network” which supports communications according to one or more higher level or lower level communications protocols such as UDP, TCP, FTP, ISDN, PPP, FDDI and others. (Specification, paras. 0013, 0023 and 0024)

In contrast, Averbuj teaches a sequence of commands included in a test algorithm for testing memory in a device. The command protocol of Averbuj is simply defines the rules for algorithms used to test memory modules. (Averbuj, Abstract, para, 0030)

There is no mention of “network traffic” in Averbuj. There is no teaching of network communications in Averbuj.

Moreover, claims 1 and 11 recite “invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command”. And claims 21, 26 and 31 recite “each protocol engine to execute a command included in one of the scripts”. In contrast, the sequencers in Averbuj receive a complete BIST algorithm and execute the algorithm one command at a time. (Averbuj, para. 0030) That is, the protocol engines as claimed receive and execute a single command from the test script or scripts, whereas the sequencers in Averbuj receive an entire algorithm. The sequencers of Averbuj are arguably similar to the claimed script interpreters, but they are also unlike the claimed script interpreters in that the sequencers execute the scripts (that is, algorithms). As such, the sequencers of Averbuj that receive and execute an entire algorithm of multiple commands do not disclose the claimed protocol engine that receives single commands.

Using the broadest reasonable interpretation of the claim term “protocol engine” in view of the specification yields the conclusion that Averbuj cannot teach a “protocol engine” as claimed.

C. Claims 2 and 12: The Combination of References Can Not Be Cited For Teaching Simulating Actions Taken by a Network User

We assert that Smith cannot be cited for disclosing that the script invoked by the protocol engine simulates the actions taken by a network user, because, as stated by the Examiner, there is no protocol engine disclosed in Smith.

In addition, we assert that the chip testing system of Averbuj cannot teach this limitation because Averbuj does not in any way involve a network or actions taken by a network user. That is, the sequencers of the memory chip testing system of Averbuj (which the Examiner asserts teach the

protocol engines) are not capable of performing actions that “simulate actions taken by a network user”.

D. Claims 5 and 15: The Combination of References Can Not Be Cited For Teaching Test Scripts That Cause Network Traffic To Be Produced

Claims 5 and 15 recite that “the test script causes network traffic to be produced”.

The Examiner asserts that Smith teaches the claimed network traffic that is produced by the protocol sequencers of Averbuj. But this cannot be, as the protocol sequencers that perform the chip testing teachings of Averbuj have no network communications capabilities whatsoever.

Because the components cited by the Examiner in the cited references are incapable of teaching a test script that causes network traffic to be produced, the references do not teach the limitations for which they are cited. As such, claims 5 and 15 are patentable over the cited references.

E. All Claims: The References Do Not Disclose The Claimed “Application Thread”

The independent claims recite an “application thread”. However, there is nothing in Smith that discloses application threads. The threads taught in Smith are traditional operating system threads.

F. Claims 21-35: The References Do Not Disclose the Claimed “User Space” and “Operating System Space”

Claims 21 through 35 recite different elements in “user space” and “operating system space”. Neither Smith nor Averbuj disclose both “user space” and “operating system space”. Both Neither Smith nor Averbuj fail to make a distinction between “user space” and “operating system space”.

G. Claims 21-25: The References Do Not Disclose the Claimed “Script Interpreter Units”, “Application Thread” and “Protocol Engines” in “User Space”

Because neither Smith nor Averbuj disclose either “user space” or “operating system space” neither can be cited for disclosing “a plurality of script interpreter units in user space”, “an application thread in user space” and “a plurality of protocol engines in user space” as claimed.

H. Claims 26-30: The References Do Not Disclose the Claimed “Script Interpreter Units and “Application Thread” in “User Space” and the Claimed “Protocol Engines” in “Operating System Space”

Because neither Smith nor Averbuj disclose either “user space” or “operating system space” neither can be cited for disclosing “a plurality of script interpreter units in user space”, “an application thread in user space” and “a plurality of protocol engines in operating system space”.

G. Claims 31-35: The References Do Not Disclose the Claimed “Script Interpreter Units” in “User Space” and the Claimed “Application Thread” and “Protocol Engines” in “Operating System Space”

Because neither Smith nor Averbuj disclose either “user space” or “operating system space” neither can be cited for disclosing “a plurality of script interpreter units in user space”, “an application thread in operating system space” and “a plurality of protocol engines in operating system space”.

CONCLUSION AND RELIEF

It is believed that all claims patentably define the subject invention over the prior art of record and are in condition for allowance. The undersigned requests that the Board overturn the rejection of all claims and hold that all of the claims of the above referenced application are allowable.

Respectfully submitted,
SoCal IP Law Group LLP



Mark A. Goldstein
Reg. No. 50,759

Date: December 22, 2008

SoCal IP Law Group LLP
310 N. Westlake Blvd., Suite 120
Westlake Village, CA 91362
Telephone: 805/230-1350
Facsimile: 805/230-1355
Email: info@socalip.com